REFERENCES

[1] M. R. Cox, *Cinderella: Three hundred and forty-five variants of Cinderella, Catskin, and Cap o'Rushes*. No. 31, Folk-lore Society, 1893.

[2] Tim Berners-Lee, "The WorldWideWeb Browser." `https://www.w3.org/People/Berners-Lee/WorldWideWeb.html`, 1990.

[3] A. Guha, C. Saftoiu, and S. Krishnamurthi, "The Essence of JavaScript," in *ECOOP 2010 - Object-Oriented Programming*, vol. 6183, pp. 126–150, 2010.

[4] M. Mulazzani, P. Reschl, M. Huber, M. Leithner, S. Schrittwieser, E. Weippl, and F. Wien, "Fast and Reliable Browser Identification With Javascript Engine Fingerprinting," in *Web 2.0 Workshop on Security and Privacy (W2SP)*, vol. 5, p. 4, Citeseer, 2013.

[5] D. Yu, A. Chander, N. Islam, and I. Serikov, "JavaScript Instrumentation for Browser Security," in *Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL*, pp. 237–249, 2007.

[6] Y. Ko, T. Rezk, and M. Serrano, "SecureJS Compiler: Portable Memory Isolation in JavaScript," in *SAC '21: The 36th ACM/SIGAPP Symposium on Applied Computing*, pp. 1265–1274, 2021.

[7] A. Haas, A. Rossberg, D. L. Schuff, B. L. Titzer, M. Holman, D. Gohman, L. Wagner, A. Zakai, and J. F. Bastien, "Bringing the Web Up to Speed With WebAssembly," in *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2017, Barcelona, Spain, June 18-23, 2017*, pp. 185–200, 2017.

[8] C. Watt, "Mechanising and Verifying the WebAssembly Specification," in *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP*, pp. 53–65, 2018.

[9] S. Narayan, T. Garfinkel, S. Lerner, H. Shacham, and D. Stefan, "Gobi: WebAssembly as a Practical Path to Library Sandboxing," *CoRR*, vol. abs/1912.02285, 2019.

[10] P. Mendki, "Evaluating Webassembly Enabled Serverless Approach for Edge Computing," in *2020 IEEE Cloud Summit*, pp. 161–166, 2020.

[11] M. Jacobsson and J. Willén, "Virtual Machine Execution for Wearables Based on WebAssembly," in *13th EAI International Conference on Body Area Networks, BODYNETS*, pp. 381–389, 2018.

[12] J. Ménétrey, M. Pasin, P. Felber, and V. Schiavoni, "WebAssembly as a Common Layer for the Cloud-Edge Continuum," in *Proceedings of the 2nd Workshop on Flexible Resource and Application Management on the Edge*, FRAME '22, p. 3–8, 2022.

[13] M. Chadha, N. Krueger, J. John, A. Jindal, M. Gerndt, and S. Benedict, "Exploring the Use of WebAssembly in HPC," in *Proceedings of the 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, PPoPP '23, p. 92–106, 2023.

[14] J. Cabrera-Arteaga, M. Monperrus, and B. Baudry, "Scalable Comparison of JavaScript V8 Bytecode Traces," in *Proceedings of the 11th ACM SIGPLAN International Workshop on Virtual Machines and Intermediate Languages, VMIL at SPLASH 2019*, pp. 22–31, 2019.

[15] NSA, "National Cyber Leap Year." `https://www.nitrd.gov/nitrdgroups/index.php?title=National_Cyber_Leap_Year`, 2021.

[16] G. Goth, "Addressing the Monoculture," *IEEE Security & Privacy*, vol. 1, no. 06, pp. 8–10, 2003.

[17] M. N. Hoque and K. A. Harras, "WebAssembly for Edge Computing: Potential and Challenges," *IEEE Communications Standards Magazine*, vol. 6, no. 4, pp. 68–73, 2022.

[18] T. Rokicki, C. Maurice, M. Botvinnik, and Y. Oren, "Port Contention Goes Portable: Port Contention Side Channels in Web Browsers," in *ASIA CCS '22: ACM Asia Conference on Computer and Communications Security*, pp. 1182–1194, 2022.

[19] S. Song, S. Park, and D. Kwon, "metaSafer: A Technique to Detect Heap Metadata Corruption in WebAssembly," *IEEE Access*, vol. 11, pp. 124887–124898, 2023.

[20] D. Lehmann, J. Kinder, and M. Pradel, "Everything Old is New Again: Binary Security of WebAssembly," in *29th USENIX Security Symposium*, pp. 217–234, 2020.

[21] Q. Stiévenart, C. D. Roover, and M. Ghafari, "Security Risks of Porting C Programs to Webassembly," in *SAC '22: The 37th ACM/SIGAPP Symposium on Applied Computing*, pp. 1713–1722, 2022.

[22] D. Genkin, L. Pachmanov, E. Tromer, and Y. Yarom, "Drive-by Key-extraction Cache Attacks from Portable Code," *IACR Cryptol. ePrint Arch.*, p. 119, 2018.

[23] G. Maisuradze and C. Rossow, "ret2spec: Speculative Execution Using Return Stack Buffers," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS*, pp. 2109–2122, 2018.

[24] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, "Thieves in the Browser: Web-based Cryptojacking in the Wild," in *Proceedings of the 14th International Conference on Availability, Reliability and Security, ARES 2019, Canterbury, UK, August 26-29, 2019*, pp. 4:1–4:10, ACM, 2019.

[25] E. Tekiner, A. Acar, A. S. Uluagac, E. Kirda, and A. A. Selçuk, "In-browser Cryptomining for Good: An Untold Story," in *IEEE International Conference on Decentralized Applications and Infrastructures, DAPPS 2021, Online Event, August 23-26, 2021*, pp. 20–29, IEEE, 2021.

[26] R. K. Konoth, E. Vineti, V. Moonsamy, M. Lindorfer, C. Kruegel, H. Bos, and G. Vigna, "MineSweeper: An In-depth Look into Drive-by Cryptocurrency Mining and Its Defense," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS*, pp. 1714–1730, 2018.

[27] A. Romano, Y. Zheng, and W. Wang, "MinerRay: Semantics-aware Analysis for Ever-evolving Cryptojacking Detection," in *35th IEEE/ACM International Conference on Automated Software Engineering, ASE 2020, Melbourne, Australia, September 21-25, 2020*, pp. 1129–1140, IEEE, 2020.

[28] F. N. Naseem, A. Aris, L. Babun, E. Tekiner, and A. S. Uluagac, "MINOS: A Lightweight Real-time Cryptojacking Detection System," in *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*, The Internet Society, 2021.

[29] W. Wang, B. Ferrell, X. Xu, K. W. Hamlen, and S. Hao, "SEISMIC: SEcure In-lined Script Monitors for Interrupting Cryptojacks," in *Computer Security - 23rd European Symposium on Research in Computer Security, ESORICS*, vol. 11099, pp. 122–142, 2018.

[30] J. D. P. Rodriguez and J. Posegga, "RAPID: Resource and API-based Detection Against In-browser Miners," in *Proceedings of the 34th Annual Computer Security Applications Conference, ACSAC 2018, San Juan, PR, USA, December 03-07, 2018*, pp. 313–326, ACM, 2018.

[31] A. Kharraz, Z. Ma, P. Murley, C. Lever, J. Mason, A. Miller, N. Borisov, M. Antonakakis, and M. Bailey, "Outguard: Detecting In-browser Covert Cryptocurrency Mining in the Wild," in *The World Wide Web Conference, WWW*, pp. 840–852, 2019.

[32] H. Okhravi, M. Rabe, T. Mayberry, W. Leonard, T. Hobson, D. Bigelow, and W. Streilein, "Survey of Cyber Moving Targets," *Massachusetts Inst of Technology Lexington Lincoln Lab, No. MIT/LL-TR-1166*, 2013.

[33] F. B. Cohen, "Operating System Protection Through Program Evolution.," *Computers & Security*, vol. 12, no. 6, pp. 565–584, 1993.

[34] S. Forrest, A. Somayaji, and D. Ackley, "Building Diverse Computer Systems," in *Proceedings. The Sixth Workshop on Hot Topics in Operating Systems (Cat. No.97TB100133)*, pp. 67–72, 1997.

[35] M. Eichin and J. Rochlis, "With microscope and tweezers: an analysis of the Internet virus of November 1988," in *Proceedings. 1989 IEEE Symposium on Security and Privacy*, pp. 326–343, 1989.

[36] J. C. Arteaga, O. F. Malivitsis, O. L. V. Pérez, B. Baudry, and M. Monperrus, "Crow: Code diversification for webassembly," in *Proceedings of MadWEB, NDSS*, 2021.

[37] J. Cabrera-Arteaga, P. Laperdrix, M. Monperrus, and B. Baudry, "Multi-variant Execution at the Edge," in *Proceedings of the 9th ACM Workshop on Moving Target Defense, MTD*, pp. 11–22, ACM, 2022.

[38] J. Cabrera-Arteaga, N. Fitzgerald, M. Monperrus, and B. Baudry, "WASM-MUTATE: Fast and Effective Binary Diversification for WebAssembly," *Computers & Security*, 2024.

[39] J. Cabrera-Arteaga, M. Monperrus, T. Toady, and B. Baudry, "WebAssembly Diversification for Malware Evasion," *Computers & Security*, vol. 131, p. 103296, 2023.

[40] J. Cabrera Arteaga, "Artificial Software Diversification for WebAssembly," No. 2022:52 in TRITA-EECS-AVL, p. 112, 2022. `https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-317331`.

[41] "Webassembly system interface." `https://github.com/WebAssembly/WASI`, 2021.

[42] D. Bryant, "WebAssembly Outside the Browser: A New Foundation for Pervasive Computing," in *Proc. of ICWE 2020*, pp. 9–12, 2020.

[43] B. Spies and M. Mock, "An Evaluation of WebAssembly in Non-web Environments," in *XLVII Latin American Computing Conference, CLEI 2021, Cartago, Costa Rica, October 25-29, 2021*, pp. 1–10, IEEE, 2021.

[44] E. Wen and G. Weber, "Wasmachine: Bring IoT up to Speed with A WebAssembly OS," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2020, Austin, TX, USA, March 23-27, 2020*, pp. 1–4, IEEE, 2020.

[45] A. Hilbig, D. Lehmann, and M. Pradel, "An Empirical Study of Real-world WebAssembly Binaries: Security, Languages, Use Cases," in *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pp. 2696–2708, 2021.

[46] Y. Yan, T. Tu, L. Zhao, Y. Zhou, and W. Wang, "Understanding the Performance of Webassembly Applications," in *Proceedings of the 21st ACM Internet Measurement Conference*, IMC '21, p. 533–549, 2021.

[47] L. Wagner, M. Mayer, A. Marino, A. S. Nezhad, H. Zwaan, and I. Malavolta, "On the Energy Consumption and Performance of WebAssembly Binaries across Programming Languages and Runtimes in IoT," in *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering, EASE 2023, Oulu, Finland, June 14-16, 2023*, pp. 72–82, ACM, 2023.

[48] B. L. Titzer, "Whose baseline compiler is it anyway?," *arXiv e-prints*, p. arXiv:2305.13241, May 2023.

[49] N. Mäkitalo, T. Mikkonen, C. Pautasso, V. Bankowski, P. Daubaris, R. Mikkola, and O. Beletski, "WebAssembly Modules as Lightweight Containers for Liquid IoT Applications," in *Proceedings of Web Engineering - 21st International Conference, ICWE*, vol. 12706, pp. 328–336, 2021.

[50] P. K. Gadepalli, S. McBride, G. Peach, L. Cherkasova, and G. Parmer, "Sledge: a Serverless-first, Light-weight Wasm Runtime for the Edge," in *Middleware '20: 21st International Middleware Conference*, pp. 265–279, 2020.

[51] N. Burow, S. A. Carr, J. Nash, P. Larsen, M. Franz, S. Brunthaler, and M. Payer, "Control-flow integrity: Precision, security, and performance," *ACM Comput. Surv.*, vol. 50, apr 2017.

[52] I. Bastys, M. Algehed, A. Sjösten, and A. Sabelfeld, "SecWasm: Information Flow Control for WebAssembly," in *Static Analysis - 29th International Symposium, SAS*, vol. 13790 of *Lecture Notes in Computer Science*, pp. 74–103, Springer, 2022.

[53] T. Brito, P. Lopes, N. Santos, and J. F. Santos, "Wasmati: An efficient static vulnerability scanner for WebAssembly," *Comput. Secur.*, vol. 118, p. 102745, 2022.

[54] F. Marques, J. Fragoso Santos, N. Santos, and P. Adão, "Concolic Execution for WebAssembly," Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[55] C. Watt, J. Renner, N. Popescu, S. Cauligi, and D. Stefan, "CT-wasm: Type-driven Secure Cryptography for the Web Ecosystem," *Proc. ACM Program. Lang.*, vol. 3, no. POPL, pp. 77:1–77:29, 2019.

[56] R. Tsoupidi, M. Balliu, and B. Baudry, "Vivienne: Relational Verification of Cryptographic Implementations in WebAssembly," in *IEEE Secure Development Conference, SecDev 2021, Atlanta, GA, USA, October 18-20, 2021*, pp. 94–102, IEEE, 2021.

[57] Q. Stiévenart and C. De Roover, "Wassail: A WebAssembly Static Analysis Library," in *Fifth International Workshop on Programming Technology for the Future Web*, 2021.

[58] F. Breitfelder, T. Roth, L. Baumgärtner, and M. Mezini, "WasmA: A Static WebAssembly Analysis Framework for Everyone," in *IEEE International Conference on Software Analysis, Evolution and Reengineering, SANER*, pp. 753–757, 2023.

[59] W. Fu, R. Lin, and D. Inge, "TaintAssembly: Taint-based Information Flow Control Tracking for WebAssembly," *CoRR*, vol. abs/1802.01050, 2018.

[60] Q. Stiévenart, D. Binkley, and C. De Roover, "Dynamic Slicing of WebAssembly Binaries," in *39th IEEE International Conference on Software Maintenance and Evolution*, IEEE, 2023.

[61] Q. Stiévenart, D. W. Binkley, and C. D. Roover, "Static Stack-preserving Intra-procedural Slicing of WebAssembly Binaries," in *44th IEEE/ACM 44th International Conference on Software Engineering, ICSE 2022, Pittsburgh, PA, USA, May 25-27, 2022*, pp. 2031–2042, ACM, 2022.

[62] D. Lehmann and M. Pradel, "Wasabi: A Framework for Dynamically Analyzing WebAssembly," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS*, pp. 1045–1058, 2019.

[63] S. Narayan, C. Disselkoen, D. Moghimi, S. Cauligi, E. Johnson, Z. Gang, A. Vahldiek-Oberwagner, R. Sahita, H. Shacham, D. M. Tullsen, and D. Stefan, "Swivel: Hardening WebAssembly against Spectre," in *30th USENIX Security Symposium, USENIX*, pp. 1433–1450, 2021.

[64] M. Kolosick, S. Narayan, E. Johnson, C. Watt, M. LeMay, D. Garg, R. Jhala, and D. Stefan, "Isolation Without Taxation: Near-Zero-cost Transitions for WebAssembly And SFI," *Proc. ACM Program. Lang.*, vol. 6, no. POPL, pp. 1–30, 2022.

[65] E. Johnson, E. Laufer, Z. Zhao, D. Gohman, S. Narayan, S. Savage, D. Stefan, and F. Brown, "WaVe: A Verifiably Secure WebAssembly Sandboxing Runtime," in *44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, May 21-25, 2023*, pp. 2940–2955, IEEE, 2023.

[66] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, "New Kid on the Web: A Study on the Prevalence of WebAssembly in the Wild," in *Detection of Intrusions and Malware, and Vulnerability Assessment - 16th International Conference, DIMVA*, vol. 11543, pp. 23–42, 2019.

[67] S. Bhansali, A. Aris, A. Acar, H. Oz, and A. S. Uluagac, "A First Look at Code Obfuscation for WebAssembly," in *WiSec '22: 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 140–145, 2022.

[68] B. Baudry and M. Monperrus, "The Multiple Facets of Software Diversity: Recent Developments in Year 2000 and Beyond," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 16:1–16:26, 2015.

[69] K. Pohl, G. Böckle, and F. van der Linden, *Software Product Line Engineering - Foundations, Principles, and Techniques.* Springer, 2005.

[70] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. C. Rinard, "Managing Performance vs. Accuracy Trade-offs With Loop Perforation," in *SIGSOFT/FSE'11 19th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-19) and ESEC'11: 13th European Software Engineering Conference (ESEC-13)*, pp. 124–134, 2011.

[71] Avizienis and Kelly, "Fault Tolerance by Design Diversity: Concepts and Experiments," *Computer*, vol. 17, no. 8, pp. 67–80, 1984.

[72] T. Y. Chen, F. Kuo, R. G. Merkel, and T. H. Tse, "Adaptive Random Testing: The ART of test case diversity," *J. Syst. Softw.*, vol. 83, no. 1, pp. 60–66, 2010.

[73] T. Jackson, *On the Design, Implications, and Effects of Implementing Software Diversity for Security.* PhD thesis, University of California, Irvine, 2012.

[74] T. Thüm, S. Apel, C. Kästner, I. Schaefer, and G. Saake, "A classification and survey of analysis strategies for software product lines," *ACM Comput. Surv.*, vol. 47, jun 2014.

[75] G. R. Lundquist, V. Mohan, and K. W. Hamlen, "Searching for Software Diversity: Attaining Artificial Diversity through Program Synthesis," in *Proceedings of the 2016 New Security Paradigms Workshop*, NSPW '16, p. 80–91, 2016.

[76] P. Koopman and J. DeVale, "Comparing the robustness of POSIX operating systems," in *Digest of Papers. Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing (Cat. No.99CB36352)*, pp. 30–37, 1999.

[77] I. Gashi, P. Popov, and L. Strigini, "Fault Diversity among Off-The-Shelf SQL Database Servers," in *Proceedings of the 2004 International Conference on Dependable Systems and Networks*, DSN '04, p. 389, 2004.

[78] J. C. Knight and N. G. Leveson, "An experimental evaluation of the assumption of independence in multiversion programming," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 1, pp. 96–109, 1986.

[79] B. Randell, "System Structure for Software Fault Tolerance," *SIGPLAN Not.*, vol. 10, p. 437–449, apr 1975.

[80] N. Harrand, *Software Diversity for Third-Party Dependencies*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2022.

[81] J. V. Cleemput, B. Coppens, and B. D. Sutter, "Compiler Mitigations for Time Attacks on Modern X86 Processors," *ACM Trans. Archit. Code Optim.*, vol. 8, no. 4, pp. 23:1–23:20, 2012.

[82] A. Homescu, S. Neisius, P. Larsen, S. Brunthaler, and M. Franz, "Profile-guided Automated Software Diversity," in *Proceedings of the 2013 IEEE/ACM International Symposium on Code Generation and Optimization, CGO 2013, Shenzhen, China, February 23-27, 2013*, pp. 23:1–23:11, IEEE Computer Society, 2013.

[83] S. Bhatkar, D. C. DuVarney, and R. Sekar, "Address Obfuscation: An Efficient Approach to Combat a Board Range of Memory Error Exploits," in *Proceedings of the USENIX Security Symposium*, 2003.

[84] S. Bhatkar and D. C. DuVarney, "Efficient Techniques for Comprehensive Protection from Memory Error Exploits," in *Proceedings of the 14th USENIX*, 2005.

[85] K. Pettis and R. C. Hansen, "Profile Guided Code Positioning," in *Proceedings of the ACM SIGPLAN'90 Conference on Programming Language Design and Implementation (PLDI)*, pp. 16–27, 1990.

[86] S. Crane, A. Homescu, S. Brunthaler, P. Larsen, and M. Franz, "Thwarting Cache Side-channel Attacks Through Dynamic Software Diversity," in *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*, The Internet Society, 2015.

[87] A. Romano, D. Lehmann, M. Pradel, and W. Wang, "Wobfuscator: Obfuscating JavaScript Malware via Opportunistic Translation to WebAssembly," in *2022 2022 IEEE Symposium on Security and Privacy (SP) (SP)*, pp. 1101–1116, may 2022.

[88] M. T. Aga and T. M. Austin, "Smokestack: Thwarting DOP Attacks with Runtime Stack Layout Randomization," in *IEEE/ACM International Symposium on Code Generation and Optimization, CGO*, pp. 26–36, 2019.

[89] S. Lee, H. Kang, J. Jang, and B. B. Kang, "SaVioR: Thwarting Stack-based Memory Safety Violations by Randomizing Stack Layout," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 4, pp. 2559–2575, 2022.

[90] Y. Younan, D. Pozza, F. Piessens, and W. Joosen, "Extended Protection against Stack Smashing Attacks without Performance Loss," in *22nd Annual Computer Security Applications Conference (ACSAC 2006), 11-15 December 2006, Miami Beach, Florida, USA*, pp. 429–438, IEEE Computer Society, 2006.

[91] Y. Xu, Y. Solihin, and X. Shen, "MERR: Improving Security of Persistent Memory Objects via Efficient Memory Exposure Reduction and Randomization," in *ASPLOS '20: Architectural Support for Programming Languages and Operating Systems*, pp. 987–1000, 2020.

[92] G. S. Kc, A. D. Keromytis, and V. Prevelakis, "Countering Code-injection Attacks With Instruction-set Randomization," in *Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS*, pp. 272–280, 2003.

[93] E. G. Barrantes, D. H. Ackley, T. S. Palmer, D. Stefanovic, and D. D. Zovi, "Randomized Instruction Set Emulation to Disrupt Binary Code Injection Attacks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS*, pp. 281–289, 2003.

[94] M. Chew and D. Song, "Mitigating Buffer Overflows by Operating System Randomization," Tech. Rep. CS-02-197, Carnegie Mellon University, 2002.

[95] D. Couroussé, T. Barry, B. Robisson, P. Jaillon, O. Potin, and J. Lanet, "Runtime Code Polymorphism as a Protection Against Side Channel Attacks," in *Proceedings of Information Security Theory and Practice - 10th IFIP WG 11.2 International Conference, WISTP*, vol. 9895, pp. 136–152, 2016.

[96] S. Cao, N. He, Y. Guo, and H. Wang, "WASMixer: Binary Obfuscation for WebAssembly," *CoRR*, vol. abs/2308.03123, 2023.

[97] C. Collberg, C. Thomborson, and D. Low, "A taxonomy of obfuscating transformations," tech. rep., Department of Computer Science, The University of Auckland, New Zealand, 1997.

[98] M. Jacob, M. H. Jakubowski, P. Naldurg, C. W. Saw, and R. Venkatesan, "The Superdiversifier: Peephole Individualization for Software Protection,"

in *Proceedings of Advances in Information and Computer Security, Third International Workshop on Security, IWSEC 2008*, vol. 5312, pp. 100–120, 2008.

[99] M. Henry, "Superoptimizer: A Look at the Smallest Program," *ACM SIGARCH Computer Architecture News*, vol. 15, pp. 122–126, Nov 1987.

[100] V. Le, M. Afshari, and Z. Su, "Compiler Validation via Equivalence Modulo Inputs," in *ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI*, pp. 216–226, 2014.

[101] B. R. Churchill, O. Padon, R. Sharma, and A. Aiken, "Semantic Program Alignment for Equivalence Checking," in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI*, pp. 1027–1040, 2019.

[102] V. Le, C. Sun, and Z. Su, "Finding Deep Compiler Bugs via Guided Stochastic Program Mutation," in *Proceedings of the 2015 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, p. 386–399, 2015.

[103] E. Schulte, Z. P. Fry, E. Fast, W. Weimer, and S. Forrest, "Software mutational robustness," vol. 15, p. 281–312, sep 2014.

[104] B. Baudry, S. Allier, and M. Monperrus, "Tailored source code transformations to synthesize computationally diverse program variants," ISSTA 2014, p. 149–159, 2014.

[105] M. Zalewski, "American Fuzzy Lop," 2017.

[106] K. Zhang, D. Wang, J. Xia, W. Y. Wang, and L. Li, "ALGO: Synthesizing Algorithmic Programs with Generated Oracle Verifiers," *CoRR*, vol. abs/2305.14591, 2023.

[107] L. de Moura and N. Bjørner, "Z3: An Efficient SMT Solver," in *Tools and Algorithms for the Construction and Analysis of Systems*, pp. 337–340, 2008.

[108] A. Abate, C. David, P. Kesseli, D. Kroening, and E. Polgreen, "Counterexample Guided Inductive Synthesis Modulo Theories," in *Proceedings of Computer Aided Verification - 30th International Conference, CAV*, vol. 10981, pp. 270–288, 2018.

[109] P. M. Phothilimthana, A. Thakur, R. Bodík, and D. Dhurjati, "Scaling up Superoptimization," in *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS*, pp. 297–310, 2016.

[110] R. El-Khalil and A. D. Keromytis, "Hydan: Hiding Information in Program Binaries," in *Information and Communications Security, 6th International Conference, ICICS*, vol. 3269, pp. 187–199, 2004.

[111] V. Singhal, A. A. Pillai, C. Saumya, M. Kulkarni, and A. Machiry, "Cornucopia : A Framework for Feedback Guided Generation of Binaries," in *37th IEEE/ACM International Conference on Automated Software Engineering, ASE 2022, Rochester, MI, USA, October 10-14, 2022*, pp. 27:1–27:13, ACM, 2022.

[112] B. Cox and D. Evans, "N-Variant Systems: A Secretless Framework for Security through Diversity," in *Proceedings of the 15th USENIX*, 2006.

[113] D. Bruschi, L. Cavallaro, and A. Lanzi, "Diversified Process replicæ for Defeating Memory Error Exploits," in *Proceedings of the 26th IEEE International Performance Computing and Communications Conference, IPCCC 2007, April 11-13, 2007, New Orleans, Louisiana, USA*, pp. 434–441, IEEE Computer Society, 2007.

[114] B. Salamat, A. Gal, T. Jackson, K. Manivannan, G. Wagner, and M. Franz, "Stopping Buffer Overflow Attacks at Run-Time: Simultaneous Multi-variant Program Execution on a Multicore Processor," tech. rep., Technical Report 07-13, School of Information and Computer Sciences, UCIrvine, 2007.

[115] L. Davi, C. Liebchen, A. Sadeghi, K. Z. Snow, and F. Monrose, "Isomeron: Code Randomization Resilient to (Just-In-Time) Return-oriented Programming," in *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*, The Internet Society, 2015.

[116] G. Agosta, A. Barenghi, G. Pelosi, and M. Scandale, "The MEET Approach: Securing Cryptographic Embedded Software Against Side Channel Attacks," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 34, no. 8, pp. 1320–1333, 2015.

[117] T. Jackson, B. Salamat, A. Homescu, K. Manivannan, G. Wagner, A. Gal, S. Brunthaler, C. Wimmer, and M. Franz, "Compiler-generated Software Diversity," in *Moving Target Defense - Creating Asymmetric Uncertainty for Cyber Threats*, vol. 54, pp. 77–98, 2011.

[118] A. Amarilli, S. Müller, D. Naccache, D. Page, P. Rauzy, and M. Tunstall, "Can Code Polymorphism Limit Information Leakage?," in *Proceedings of Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication - 5th IFIP WG 11.2 International Workshop, WISTP*, vol. 6633, pp. 1–21, 2011.

[119] A. Voulimeneas, D. Song, P. Larsen, M. Franz, and S. Volckaert, "dMVX: Secure and Efficient Multi-variant Execution in a Distributed Setting," in *EuroSec '21: Proceedings of the 14th European Workshop on Systems Security, Virtual Event / Edinburgh, Scotland, UK, April 26, 2021*, pp. 41–47, ACM, 2021.

[120] B. De Sutter, B. Anckaert, J. Geiregat, D. Chanet, and K. De Bosschere, "Instruction Set Limitation in Support of Software Diversity," pp. 152–165, 2009.

[121] R. Tsoupidi, R. C. Lozano, and B. Baudry, "Constraint-based Diversification of JOP Gadgets," *J. Artif. Intell. Res.*, vol. 72, pp. 1471–1505, 2021.

[122] J. Falleri, F. Morandat, X. Blanc, M. Martinez, and M. Monperrus, "Fine-grained and Accurate Source Code Differencing," in *ACM/IEEE International Conference on Automated Software Engineering, ASE '14*, pp. 313–324, 2014.

[123] S. Banescu, C. Collberg, and A. Pretschner, "Predicting the Resilience of Obfuscated Code Against Symbolic Execution Attacks via Machine Learning," in *26th USENIX Security Symposium (USENIX Security 17)*, pp. 661–678, Aug. 2017.

[124] H. Bostani and V. Moonsamy, "EvadeDroid: A Practical Evasion Attack on Machine Learning for Black-box Android Malware Detection," *CoRR*, vol. abs/2110.03301, 2021.

[125] D. D. Yao, X. Shu, L. Cheng, and S. J. Stolfo, *Anomaly Detection as a Service: Challenges, Advances, and Opportunities*. Synthesis Lectures on Information Security, Privacy, and Trust, Morgan & Claypool Publishers, 2017.

[126] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion Detection Using Sequences of System Calls," *J. Comput. Secur.*, vol. 6, no. 3, pp. 151–180, 1998.

[127] Y. Fang, C. Huang, L. Liu, and M. Xue, "Research on Malicious JavaScript Detection Technology Based on LSTM," *IEEE Access*, vol. 6, pp. 59118–59125, 2018.

[128] E. Johnson, D. Thien, Y. Alhessi, S. Narayan, F. Brown, S. Lerner, T. McMullen, S. Savage, and D. Stefan, ", : SFI safety for native-compiled Wasm," *Network and Distributed Systems Security (NDSS) Symposium*, 2021.

[129] F. Cohen, "Computer Viruses," in *Proceedings of the 7th DoD/NBS Computer Security Conference 1984*, pp. 240–263, 1986.

[130] R. L. Castro, C. Schmitt, and G. D. Rodosek, "ARMED: How Automatic Malware Modifications Can Evade Static Detection?," in *2019 5th International Conference on Information Management (ICIM)*, pp. 20–27, 2019.

[131] R. L. Castro, C. Schmitt, and G. Dreo, "AIMED: Evolving Malware with Genetic Programming to Evade Detection," in *18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications / 13th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE 2019, Rotorua, New Zealand, August 5-8, 2019*, pp. 240–247, IEEE, 2019.

[132] W. Wang, Y. Zheng, X. Xing, Y. Kwon, X. Zhang, and P. Eugster, "WebRanz: Web Page Randomization for Better Advertisement Delivery and Web-Bot Prevention," FSE 2016, p. 205–216, 2016.

[133] H. Aghakhani, F. Gritti, F. Mecca, M. Lindorfer, S. Ortolani, D. Balzarotti, G. Vigna, and C. Kruegel, "When Malware is Packin' Heat; Limits of Machine Learning Classifiers Based on Static Analysis Features," in *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*, The Internet Society, 2020.

[134] M. W. J. Chua and V. Balachandran, "Effectiveness of Android Obfuscation on Evading Anti-malware," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, CODASPY*, pp. 143–145, 2018.

[135] P. Dasgupta and Z. Osman, "A Comparison of State-of-the-art Techniques for Generating Adversarial Malware Binaries," *CoRR*, vol. abs/2111.11487, 2021.

[136] G. Lu and S. K. Debray, "Weaknesses in Defenses against Web-borne Malware," in *Proceedings of Detection of Intrusions and Malware, and Vulnerability Assessment - 10th International Conference, DIMVA 2013*, vol. 7967, pp. 139–149, Springer, 2013.

[137] M. Payer, "Embracing the New Threat: Towards Automatically Self-diversifying Malware," in *Proceedings of The Symposium on Security for Asia Network*, pp. 1–5, 2014.

[138] N. Loose, F. Mächtle, C. Pott, V. Bezsmertnyi, and T. Eisenbarth, "Madvex: Instrumentation-based Adversarial Attacks on Machine Learning Malware Detection," in *Detection of Intrusions and Malware, and Vulnerability Assessment - 20th International Conference, DIMVA 2023*, vol. 13959 of *Lecture Notes in Computer Science*, pp. 69–88, 2023.

[139] A. V. Aho, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques, and Tools*, ch. 1, pp. 28–31. 1986.

[140] R. Sasnauskas, Y. Chen, P. Collingbourne, J. Ketema, J. Taneja, and J. Regehr, "Souper: A Synthesizing Superoptimizer," *CoRR*, vol. abs/1711.04422, 2017.

[141] B. G. Ryder, "Constructing the Call Graph of a Program," *IEEE Transactions on Software Engineering*, no. 3, pp. 216–226, 1979.

[142] S. Narayan, C. Disselkoen, D. Moghimi, S. Cauligi, E. Johnson, Z. Gang, A. Vahldiek-Oberwagner, R. Sahita, H. Shacham, D. M. Tullsen, and D. Stefan, "Swivel: Hardening WebAssembly against Spectre," in *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pp. 1433–1450, 2021.

[143] M. Willsey, C. Nandi, Y. R. Wang, O. Flatt, Z. Tatlock, and P. Panchekha, "Egg: Fast and Extensible Equality Saturation," *Proc. ACM Program. Lang.*, vol. 5, no. POPL, pp. 1–29, 2021.

[144] "Stop a wasm compiler bug before it becomes a problem | fastly." `https://www.fastly.com/blog/defense-in-depth-stopping-a-wasm-compiler-bug-before-it-became-a-problem`, 2021.

[145] D. Cao, R. Kunkel, C. Nandi, M. Willsey, Z. Tatlock, and N. Polikarpova, "babble: Learning Better Abstractions with E-Graphs and Anti-unification," *Proc. ACM Program. Lang.*, vol. 7, no. POPL, pp. 396–424, 2023.

[146] R. Tate, M. Stepp, Z. Tatlock, and S. Lerner, "Equality Saturation: A New Approach to Optimization," in *Proceedings of the 36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL*, pp. 264–276, 2009.

[147] T. D. Morgan and J. W. Morgan, "Web Timing Attacks Made Practical," *Black Hat*, 2015.

[148] T. Schnitzler, K. Kohls, E. Bitsikas, and C. Pöpper, "Hope of Delivery: Extracting User Locations From Mobile Instant Messengers," in *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*, The Internet Society, 2023.

[149] Mozilla, "Protections Against Fingerprinting and Cryptocurrency Mining Available in Firefox Nightly and Beta ," 2019.

[150] F. Cohen, "Computer Viruses: Theory and Experiments," *Comput. Secur.*, vol. 6, no. 1, pp. 22–35, 1987.

[151] P. Kocher, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, and Y. Yarom, "Spectre Attacks: Exploiting Speculative Execution," *meltdownattack.com*, 2018.

[152] M. Schwarz, C. Maurice, D. Gruss, and S. Mangard, "Fantastic Timers and Where to Find Them: High-resolution Microarchitectural Attacks in JavaScript," in *Financial Cryptography and Data Security - 21st International Conference, FC*, vol. 10322, pp. 247–267, 2017.

[153] G. J. Duck, X. Gao, and A. Roychoudhury, "Binary Rewriting Without Control Flow Recovery," in *Proceedings of the 41st ACM SIGPLAN International Conference on Programming Language Design and Implementation, PLDI*, pp. 151–163, 2020.

[154] A. Nicholson, Q. Stiévenart, A. Mazidi, and M. Ghafari, "Wasmizer: Curating WebAssembly-driven Projects on GitHub," in *2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR)*, pp. 130–141, 2023.

[155] T. Y. Zhuo, Z. Yang, Z. Sun, Y. Wang, L. Li, X. Du, Z. Xing, and D. Lo, "Source Code Data Augmentation for Deep Learning: A Survey," *arXiv e-prints*, p. arXiv:2305.19915, May 2023.

[156] S. Srikant, S. Liu, T. Mitrovska, S. Chang, Q. Fan, G. Zhang, and U. O'Reilly, "Generating Adversarial Computer Programs using Optimized Obfuscations," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[157] H. Ye, M. Martinez, X. Luo, T. Zhang, and M. Monperrus, "SelfAPR: Self-supervised Program Repair with Test Execution Diagnostics," in *37th IEEE/ACM International Conference on Automated Software Engineering, ASE 2022, Rochester, MI, USA, October 10-14, 2022*, pp. 92:1–92:13, ACM, 2022.

[158] W. Zhang, S. Guo, H. Zhang, Y. Sui, Y. Xue, and Y. Xu, "Challenging Machine Learning-based Clone Detectors via Semantic-preserving Code Transformations," *IEEE Trans. Software Eng.*, vol. 49, no. 5, pp. 3052–3070, 2023.

[159] H. Li, X. Zhou, L. A. Tuan, and C. Miao, "Rethinking Negative Pairs in Code Search," *arXiv preprint arXiv:2310.08069*, 2023.

[160] J. D. Seideman, *Transformation and Abstraction to Aid Comparison of Binary Executables Across Compilation Environments.* PhD thesis, City University of New York, 2023.

[161] H. Huang, A. M. Youssef, and M. Debbabi, "BinSequence: Fast, Accurate and Scalable Binary Code Reuse Detection," *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017.

[162] J. Jang, A. Agrawal, and D. Brumley, "ReDeBug: Finding Unpatched Code Clones in Entire OS Distributions," in *2012 IEEE Symposium on Security and Privacy*, pp. 48–62, 2012.

[163] H. Jang, K. Yang, G. Lee, Y. Na, J. D. Seideman, S. Luo, H. Lee, and S. Dietrich, "QuickBCC: Quick and Scalable Binary Vulnerable Code Clone Detection," in *ICT Systems Security and Privacy Protection*, pp. 66–82, 2021.

Part II

Included papers

WEBASSEMBLY DIVERSIFICATION FOR MALWARE EVASION

Javier Cabrera-Arteaga, Tim Toady, Martin Monperrus, Benoit Baudry
Computers & Security, Volume 131, 2023

https://www.sciencedirect.com/science/article/pii/S0167404823002067

WASM-MUTATE: FAST AND EFFECTIVE BINARY DIVERSIFICATION FOR WEBASSEMBLY

Javier Cabrera-Arteaga, Nick Fitzgerald, Martin Monperrus, Benoit Baudry
Submitted to Computers & Security, 2024

https://www.sciencedirect.com/science/article/pii/S0167404824000324

CROW: CODE DIVERSIFICATION FOR WEBASSEMBLY

Javier Cabrera-Arteaga, Orestis Floros, Oscar Vera-Pérez, Benoit Baudry, Martin Monperrus
Network and Distributed System Security Symposium (NDSS 2021), Workshop on Measurements, Attacks, and Defenses for the Web

https://doi.org/10.14722/madweb.2021.23004

MULTI-VARIANT EXECUTION AT THE EDGE

Javier Cabrera-Arteaga, Pierre Laperdrix, Martin Monperrus, Benoit Baudry
Conference on Computer and Communications Security (CCS 2022), Workshop on Moving Target Defense (MTD)

https://dl.acm.org/doi/abs/10.1145/3560828.3564007

SUPEROPTIMIZATION OF WEBASSEMBLY BYTECODE

Javier Cabrera-Arteaga, Shrinish Donde, Jian Gu, Orestis Floros, Lucas Satabin, Benoit Baudry, Martin Monperrus
Conference Companion of the 4th International Conference on Art, Science, and Engineering of Programming (Programming 2021), MoreVMs

https://doi.org/10.1145/3397537.3397567

SCALABLE COMPARISON OF JAVASCRIPT V8 BYTECODE TRACES

Javier Cabrera-Arteaga, Martin Monperrus, Benoit Baudry
11th ACM SIGPLAN International Workshop on Virtual Machines and Intermediate Languages (SPLASH 2019)

https://doi.org/10.1145/3358504.3361228